Open Source Publishing

Design Tools For Designers
« Kaffeesatz + Tagesschrift
You need to copy to understand »

If the design thinking is correct, the tools should be irrelevant

(Type) designer **Pedro Amado** is amongst many other things initiator of TypeForge, a website dedicated to the development of 'collaborative type' with open source tools. While working as design technician at FBAUP, he is about to finish a MA with a paper on collaborative methods for the creation of art and design projects. When I e-mailed him about open font design and how he sees that developing, he responded with a list of useful links, but also with:

> "Developing design teaching based on open source is one of my goals, because I think that is the future of education."

This text is based on the conversation about design, teaching and software that followed.

You told me you are employed as 'design technician'… what does that mean?

It means that I provide assistance to teachers and students in the Design Department. I implemented scanning/printing facilities for example, and currently I develop and give workshops on Digital Technologies – software is a BIG issue for me right now!

Linux and Open Source Software are slowly entering the design spaces of our school. For me it has been a 'battle' to find space for these tools. I mean - we could migrate completely to OSS tools, but it's a slow progress. Mainly because people (students) need (and want) to be trained in the same commercial applications as the ones they will encounter in their professional life.

How did Linux enter the design lab? How did that start?

It started with a personal curiosity, but also for economical reasons. Our school can't afford to acquire all the software licenses we'd like. For example, we can't justify to pay approx. 100 x 10 € licenses, just to implement the educational version of Fontlab on some of our computers; especially because this package is only used by a part of our second year design students. You can image what the total budget will be with all the other needs…

I personally believe that we can find everything we need on the web. It's a matter of searching long enough! So this is how I was very happy to find Fontforge. An open source tool that is solid enough to use in education and can produce (as far as I have been able to test) almost professional results in font development.

At first I couldn't grasp how to use it under X on Windows, so one day I set out to try and do it on Linux… and one thing lead to another…

What got you into using OSS? Was it all one thing leading to another?

Uau… can't remember… I believe it had to do with my first experiences on line; I don't think I knew the concept before 2000. I mean I've started using the web (IRC and basic browsing) in 1999, but I think it had to do with the search of newer and better tools…

I think I also started to get into it around that time. But I think I was more interested in copyleft though, than in software.

Oh… (blush) not me… I got into it definitely for the 'free beer' aspect!

By 2004 I started using DTP applications on Linux (still in my own time) and began to think that these tools could be used in an educational context, if not professionally. In the beginning of 2006 I presented a study to the coordinator of the Design Department at FBAUP, in which I proposed to start implementing Open Source tools as an alternative to the tools we were missing. Blender for 3D animation, FontForge for type design, Processing for interactive/graphic programming and others as a complement to proprietary packages: The Gimp, Scribus and Inkscape to name the most important ones.

I ran into some technical problems that I hope will be sorted out soon; one of the strategies is to run these software packages on a migration basis - as the older computers in our lab won't be able to run MacOS 10.4+, we'll start converting them to Linux.

I wanted to ask you about the relation between software and design. To me, economy, working process, but also aesthetics are a product of software, and at the same time software itself is shaped through use. I think the borders between software and design are not so strictly drawn.

It's funny you put things in that perspective. I couldn't agree more. Nevertheless I think that design thinking prevails (or it should) as it must come first when approaching problems. If the design thinking is correct, the tools used should be irrelevant. I say 'should' because in a perfect environment we could work within a team where all tools (software/hardware) are mastered. Rarely this happens, so much of our design thinking is still influenced by what we can actually produce.

Do you mean to say that "what we can think is influenced by what we can make"? This would work for me! But often when tools are mastered, they disappear in the background and in my opinion that can become a problem.

I'm not sure if I follow your point. I agree with "the border between design and software is not so strict" nevertheless, I don't agree with "economy, process and aesthetics are a product of software". As you've come to say what we think is influenced by what we can make… this is an outside observation…

> A technique is produced inside a culture, therefore one's society is conditioned by it's techniques. Conditioned, not determined" (LÉVY, 2000)

Design, like economics and software, is a product of culture. Or is it the other way around? The fact is that we can't really tell what comes first. Culture is defined by and defines technology. Therefore it's more or less simple to accept that software determines (and is determined) by it's use. This is an intricate process… it kind of goes roundabout on itself…

And where does design fit in in your opinion? Or more precisely: designers?

Design is a cultural aspect. Therefore it does not escape this logic. Using a practical sta ndpoint: Design is a product of economics and technology. Nevertheless the best design practices (or at least the one's that have endured the test of time) and the most renowned designers are the one's that can escape the the economic and technological boundaries.

The best design practices are the ones that are not products of economics and technology… t hey are kind of approaching a universal design status (if one exists). of course… it's very theoretical, and optimistic… but it should be like this… othe rwise we'll stop looking for better or newer solutions, and we'll stop pushing boundaries and design as technology and other areas will stagnate.

On the other hand, there is a special 'school' of thought manifested through some of the Po rtuguese Design Association members, saying that the design process should lead the process of technological development. Henrique Cayate (I think it w as in November last year) said that "design should lead the way to economy and technology in society." I think this is a bit far fetched…

Do you think software defines form and/or content? How is software related to design processes?

I think these are the essential questions related to the use of OSS. Can we think about wha t we can make without thinking about process? I believe that in design processes, as in design teaching, concepts should be separated from techniques or so ftware as much as possible.

To me, exactly because techniques and software are intertwined, software matters and should offer space for thinking (software should therefore n ot be separated from design).

You could also say: design becomes exceptionally strong when it makes use of its context, and responds to it in an intelligent way. Or maybe I di d not understand what you meant by being "a product of". To me that is not necessarily a negative point.

Well… yes… that could be a definition of good design, I guess.

I think that as a cultural produce, techniques can't determine society. It can and will inf luence it, but at the same time it will also just happen. When we talk about Design and Software I see the same principle reflected. Design being the "cultur e" or society and software being the tools or techniques that are developed to be used by designers. So this is much the same as "which came first? The c hicken, or the egg?" Looking at it from a designers (not a software developers) point of view, the tools we use will always condition our output. Neve rtheless I think it's our role as users to push tools further and let developers know what we want to do with them. Whether we do animation on Photoshop, or print graphics on Flash that's our responsibility. We have to use our tools in a responsible way. Knowing that the use we make of them will event ually come back at us. It's a kind of responsible feedback.

Using Linux in a design environment is not an obvious choice. Most designers are practically married to their Adobe Suite. How come it is enterin g your school after all?

Very slowly! Linux is finally becoming valuable for Design/DTP area as it has been for long on the Internet/Web and programming areas. But you can't expect The Gimp to surpass Photoshop. At least not in the next few years. And this is the r eality. If we can, we must train our students to use the best tools available. Ideally all tools available, so they won't have problems when faced with a tool professionally.

The big question is still, how we besides teaching students theory and design processes (wi th the help of free tools), help them to become professionals. We also have to teach them how to survive a professional relationship with professional too ls like the Adobe Suite. As I am certain that Linux and OSS (or FLOSS) will be part of education's future, I am certain of it's coexistence along side with commercial software like Adobe's. It's only a matter of time. Being certain of this, the essential question is: How will we manage to work parallel in both commercial and free worlds?

Do you think it is at all possible to 'survive' on other tools than the ones Adobe offers?

well… I seem not to be able to dedicate myself entirely to these new tools…

To depend solely on OSS tools… I think that is not possible, at least not at this moment. B ut now is the time to take these OSS tools and start to teach with them. They must be implemented in our schools. I am certain that sooner or later this will be common practice throughout European schools.

Can you explain a bit more, what you mean by 'real world'?

Being a professional graphic designer is what we call the 'real world' in our school. I mea n, having to work full time doing illustration, corporate identity, graphic design etc. to make a living - deliver on time to clients and make a prof it to pay the bills by the end of the month!

Do you think OSS can/should be taught differently? It seems self-teaching is built in to these tools and the community around it. It means you le arn to teach others in fact … that you actually have to leave the concept of 'mastering' behind?

I agree. The great thing about Linux is precisely that - as it is developed by users and fo r users - it is developing a sense of community around it, a sense of "given enough eyeballs, someone will figure it out"

Well, that does not always work, but most of the time…

I believe that using open source tools is perfect to teach, especially first year students. Almost no one really understands what the commands behind the menus of Photoshop mean, at least not the people I've seen in my workshops. I guess The Gim p won't resolve this matter, but it will help them think about what they are doing to digital images. Especially when they have to use unfamiliar so ftware.

You first have to teach the design process and then the tool can be taught correctly, other wise you'll just be teaching habits or tricks. As I said before, as long as design prevails and not the tool/technique, and you teach the concepts behind the t ools in the right way, people will adapt seamlessly to new tools, and the interface will become invisible!

Do you think this means you will need to restructure the curriculum? I imagine a class in bugreporting… or getting help on line…

mmhh… that could be interesting. I've never thought about it in that way. I've always seen bugreporting and other community driven activities as part of the individual aspect of working with these tools… but basically you are suggesting to impl ement an 'open source civic behavior class' or something like that?

Ehm… Yes! I think you need to learn that you own your tools, meaning you need to take care of them (ie: if something does not work, report) but a t the

same time you can open them up and get under the hood… change something small or something big. You also need to learn that you can expect to get help from other people than your tutor… and that you can teach someone else.

The aspect of taking responsibility, this has to be cultivated - a responsible use of these tools. About changing things under the hood… well this I think it will be more difficult. I think there is barely space to educate people to hack their own tools let alone getting under the hood and modifying them.

But you are right that under the OSS communication model, the peer review model of analysis, communication is getting less and less hierarchical. You don't have to be an expert to develop new or powerful tools or other things… A peer-review model assumes that you just need to be clever and willing to work with others. As long as you treat your collaborators as peers, whether or not they are more or less advanced than you, this will motivate them to work harder. You should not disregard their suggestions and reward them with the implementations (or critics) of their work.

How does that model become a reality in teaching? How can you practice this?

Well… for example use public communication/distribution platforms (like an expanded web forum) inside school, or available on the Internet; blog updates and suggestions constantly; keep a repository of files; encourage the use of real time communication technologies… as you might have noticed is almost the formula used in e-learning solutions.

And also often an argument for cutting down on teaching hours.

That actually is and isn't true. You can and will (almost certainly) have less and less traditional classes, but if the teachers and tutors are dedicated, they will be more available than ever! This will mean that students and teachers will be working together in a more informal relationship. But it can also provoke an invasion of the personal space of teachers…

It is hard to put a border when you are that much involved. I am just thinking how you could use the community around Open Source Software to help out. I mean… if the on line teaching tools would be open to others outside the school too, this would be the advantage. It would also mean that as a school, you contribute to the public domain with your classes and courses.

That is another question. I think schools should contribute to public domain knowledge. Right now I am not sharing any of the knowledge about implementing OSS on a school like ours with the community. But if all goes well I'll have this working by December 2006. I'm working on a website where I can post the handbooks for workshops and other useful resources.

I am really curious about your experiences. However convinced I am of the necessity to do it, I don't think it is easy to open education up to the public, especially not for undergraduate education.

I do have my doubts too. If you look at it on a commercial perspective, students are paying for their education… should we share the same content to everyone? Will other people explore these resources in a wrong way? Will it really contribute to the rest of the community? What about profit? Can we afford to give this knowledge away for free, I mean, as a school this is almost our only source of income? Will the prestige gained, be worth the possible loss? These are important questions that I need to think more about.

OK, I will be back with you in 6 month to find out more!

My last question… why would you invest time and energy in OSS when you think good designers should escape economical and technological boundaries?

If we invest energy on OSS tools now, we'll have the advantage of already being savvy by the time they become widely accepted. The worst case scenario would be that you've wasted time perfecting your skills or learned a new tool that didn't become a standard… How many times have we done this already in our life? In any way, we need to learn concepts behind the tools, learn new and different tools, even unnecessary ones in order to broaden our knowledge base – this will eventually help us think 'out of the box' and hopefully push boundaries further [not so much as escaping them].

For me OSS and its movement have reached a maturity level that can prove it's own worth in society. Just see Firefox - when it reached general user acceptance level (aka 'project maturity' or 'development state'), they started to compete directly with MS Internet Explorer. This will happen with the rest (at least that's what I believe). It's a matter of quality and doing the correct broadcast to the general public.

Linux started almost as a personal project and now it's a powerhouse in programming or web environments. Maybe because these are areas that require constant software and hardware attention it became an obvious and successful choice. People just modified it as they needed it done. Couldn't this be done as effectively (or better) with commercial solutions? Of course. But could people develop personalized solutions to specific problems in their own time frame? Probably not…

But it means that the people involved are, or can resource to, computer experts. What about the application of these ideas to other areas? The justice department of the Portuguese government (Ministério da Justiça) is for example currently undergoing a massive informatics (as in the tools used) change – they are slowly migrating their working platform to an Open Source Linux distribution – Caixa Mágica (although it's maintained and given assistance by a commercial enterprise by the same name). By doing this, they'll cut costs dramatically and will still be able to work with equivalent productivity (one hopes: better!). The other example is well known. The Spanish region of Estremadura looked for a way to cut costs on the implementation of information technologies in their school system and developed their own Linux Distro called Linex – it aggregates the software bundle they need, and best of all has been developed and constantly tweaked by them.

Now Linux is becoming more accessible for users without technical training, and is in a WYSIWYG state of development, I really believe we should start using it seriously so we can try and test it and learn how we can use in in our everyday life (for me this process has already started…).

People aren't stupid. They're just 'change resistant'. One of the aspects I think that will get peoples' attention will be that a 'free beer' is as good as a commercial one.

Posted by Femke on Sunday, August 6th, 2006 in: Education, Conversations. You can Comment, or trackback from your own site.

One Response to "If the design thinking is correct, the tools should be irrelevant"

1. *Manuela Amado* Says:
   September 12th, 2006 at 6:01 pm

   Então o Linux já devia ser coloado nos computadores como o MS-Dos?

Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

Submit Comment

Entries (RSS) / Comments (RSS)